

Pour un
monde connecté



 MarathonTP[©]

\\Référence du protocole V1.1

 **INERTIA**
SYSTEMES

\\Copyrights

Inertia Systèmes
73 rue des Colombes
Ange-Gardien, Québec
J0E 1E0
<http://inertiasystemes.com>
<http://marathontp.info>

\\Licence

Attribution - Pas de Modification 4.0 International



<http://creativecommons.org/licenses/by-nd/4.0/deed.fr>

\\Processus de révision

La spécification du protocole MarathonTP n'est pas statique. Celle-ci est appelée à évoluer avec le temps. Pour uniformiser l'évolution du protocole, il est de la responsabilité d'Inertia Systèmes d'en approuver les modifications. Tout utilisateur du protocole peut proposer une révision à Inertia Systèmes par les moyens de communications disponibles dans la section « Copyrights ». Sur acceptation de la révision, Inertia Systèmes s'engage à fournir à la communauté une version mise à jour du présent document.

\\Avertissements

Ce manuel de référence est à titre informatif uniquement et est fourni « tel quel ». Les informations énoncées dans le présent document sont présentées comme un guide et non comme un processus étape par étape. Nous recommandons fortement que vous engagiez une expertise supplémentaire afin d'évaluer davantage les conditions requises pour votre environnement spécifique.

INERTIA SYSTÈMES SE RÉSERVE LE DROIT D'APPORTER DES MODIFICATIONS AUX SPÉCIFICATIONS DU PROTOCOLE À TOUT MOMENT SANS PRÉAVIS.

LES UTILISATEURS DOIVENT ASSUMER L'ENTIÈRE RESPONSABILITÉ POUR L'APPLICATION DES RÈGLES MENTIONNÉS DANS LES PRÉSENTES. INERTIA SYSTÈMES N'ASSUME AUCUNE RESPONSABILITÉ, ET EXCLUT TOUTE GARANTIE, EXPRESSE OU IMPLICITE, RELATIVE À L'APPLICATION DES RÈGLES DANS LES PRÉSENTES.

L'obtention de ce document ne vous donne aucun droit de licence pour les brevets, marques, droits d'auteur ou autre propriété intellectuelle.

\\Table des matières

| | |
|--|----|
| \\Copyrights..... | 2 |
| \\Licence | 3 |
| \\Processus de révision | 4 |
| \\Avertissements | 5 |
| \\Liste des figures | 8 |
| \\Liste des tableaux | 9 |
| 1 \\Introduction | 10 |
| 2 \\Types de donnée | 11 |
| 2.1 Représentation des nombres..... | 12 |
| 2.2 Notes sur les chaînes de texte | 12 |
| 3 \\Structure de message | 13 |
| 3.1 Définition des éléments..... | 13 |
| 3.1.1 Les marqueurs de début et de fin | 13 |
| 3.1.2 Les séparateurs de champs..... | 13 |
| 3.1.3 Le descripteur | 14 |
| 3.1.4 La charge utile | 15 |
| 4 \\Commandes MarathonTP..... | 16 |
| 4.1 Lecture de variable | 16 |
| 4.1.1 Format de la requête | 16 |
| 4.1.2 Format de réponse..... | 17 |
| 4.2 Écriture de variable..... | 18 |
| 4.2.1 Format de la requête | 18 |
| 4.2.2 Format de la réponse | 19 |
| 4.3 La Découverte..... | 19 |
| 4.3.1 Format de la requête | 20 |
| 4.3.2 Format de la réponse | 20 |
| 4.4 Codes de réponse..... | 21 |
| 5 \\Organigramme des message | 22 |
| 5.1 Émission d'un message | 22 |
| 5.2 Réception d'un message..... | 23 |
| 5.3 Recyclage des messages..... | 24 |

| | | |
|--------|---|-----------|
| 6 | \\Contrôle de congestion..... | 25 |
| 6.1 | Le régulateur d'envoi | 25 |
| 6.2 | Recul Exponentiel..... | 25 |
| 7 | \\Sécurité..... | 27 |
| 7.1 | Aucune Sécurité | 27 |
| 7.2 | XTEA (eXtended TEA) | 27 |
| 7.3 | Sécurité avancée | 28 |
| 8 | \\Listes d'échange..... | 29 |
| 8.1 | Précision des index réservés | 29 |
| 8.1.1 | Index 0 : Ping | 30 |
| 8.1.2 | Index 1 : Device Serial | 30 |
| 8.1.3 | Index 2 : Device IS Identifier | 30 |
| 8.1.4 | Index 3 : Security Mode | 30 |
| 8.1.5 | Index 10 : Sended Count | 30 |
| 8.1.6 | Index 11 : Received Count | 30 |
| 8.1.7 | Index 12 : Failed Count | 31 |
| 8.1.8 | Index 13 : Retried Count | 31 |
| 8.1.9 | Index 14 : Successful Per Second | 31 |
| 8.1.10 | Index 15 : Max Retransmit Interval | 31 |
| 8.1.11 | Index 16 : Max Retry Attempt | 31 |
| 8.1.12 | Index 17 : TimeOut | 31 |
| 9 | \\Conformité du protocole..... | 32 |
| 9.1 | Port de communication UDP | 32 |
| 9.2 | Types de données..... | 32 |
| 9.3 | Taille des paquets | 32 |

\\Liste des figures

| | |
|---|----|
| Figure 1 : Modèle OSI | 10 |
| Figure 2 : Format d'un paquet MarathonTP..... | 13 |
| Figure 3 : Liste d'objets triples..... | 15 |
| Figure 4 : Organigramme d'émission..... | 22 |
| Figure 5 : Organigramme de réception..... | 23 |
| Figure 6 : Organigramme de recyclage..... | 24 |

\\Liste des tableaux

| | |
|--|----|
| Tableau 1 : Type de données du protocole | 11 |
| Tableau 2 : Caractères réservés | 12 |
| Tableau 3 : Champs de descripteur | 14 |
| Tableau 4 : Éléments d'une requête de lecture | 16 |
| Tableau 5 : Éléments d'une réponse de lecture | 17 |
| Tableau 6 : Éléments d'une requête d'écriture | 18 |
| Tableau 7 : Éléments d'une réponse d'écriture | 19 |
| Tableau 8 : Éléments d'une requête de Découverte | 20 |
| Tableau 9 : Éléments d'une réponse de Découverte | 20 |
| Tableau 10 : Codes d'erreur MarathonTP | 21 |
| Tableau 11 : Modes de sécurité | 27 |
| Tableau 12 : Répartition des plages d'index. | 29 |

1 \Introduction

MarathonTP est un protocole de communication machine à machine (M2M). Il est conçu pour être simple d'implémentation et pouvant s'adapter à une multitude de plateforme capable de gérer des chaînes de caractère. Il permet l'échange fiable de message entre dispositifs de ressources limitées. Ces dispositifs utilisent souvent des microcontrôleurs 8 à 32 bits bon marché.

MarathonTP se distingue par le fait qu'il est dit « humainement lisible ». En effet, les messages transmis sont en fait des chaînes de caractère de format [UTF-8](#). Ainsi, lors de son implémentation, il est aisé de confirmer la structure des messages transmis avec des outils d'analyse réseau comme [WireShark](#).

Selon le modèle OSI (Open Source Interconnection), MarathonTP est un protocole de la couche 7, soit la couche d'application. Du point de vue du modèle OSI, la couche d'application est responsable de la représentation des données à l'utilisateur, de leur codage et du contrôle du dialogue entre les systèmes. MarathonTP défini à la fois une structure standardisé de message ainsi qu'une méthodologie de transfert fiable. MarathonTP peut s'adapter à une panoplie de protocole de transport. Citons par exemple RS-232, UDP/IP, etc. Ce document ne couvre pas l'établissement des connections entre les machines.

Le nom du protocole a été choisi pour décrire les buts recherchés lors de son développement. Le marathon représente une épreuve d'endurance et l'acronyme « TP » signifie « Transport Protocol ». MarathonTP est donc un protocole de transport d'information fiable et durable.



Figure 1 : Modèle OSI

2 \\Types de données

Le protocole prévoit l'échange d'information sous plusieurs types. À chacun de ces types correspond donc un identificateur unique. Celui-ci permet à l'interpréteur du message de convertir la chaîne de caractère reçu dans le bon format. Le tableau suivant fait la synthèse des types que le protocole supporte :

| Type de donnée | Description | Identificateur | Valeurs possibles |
|----------------|---|----------------|--|
| BOOLEAN | Contient des valeurs qui ne peuvent être que vrai ou faux. Les mots clés True et False correspondent aux deux états de variables booléennes. | « Bo » | « True » « False » |
| INTEGER | Contient des entiers signés de 32 bits (4 octets). | « In » | Toutes valeurs entières entre : « -2147483648 » et « 2147483647 ». |
| SHORT | Contient des entiers signés de 16 bits (2 octets). | « Sh » | Toutes valeurs entières entre : « -32768 » et « 32767 ». |
| USHORT | Contient des entiers non signés de 16 bits (2 octets). | « USh » | Toutes valeurs entières entre : « 0 » et « 65535 ». |
| LONG | Contient des entiers signés de 64 bits (8 octets). | « Lo » | Toutes valeurs entières entre : « -922337236854775808 » et « 922337236854775807 ». |
| SINGLE | Contient des nombres signés IEEE 32 bits à virgule flottante de simple précision (4 octets). Les nombres en simple précision stockent une approximation d'un nombre réel. | « Si » | Pour les nombres négatifs : de « - 3,4028235E+38 » à « - 1,401298E-45 ». Pour les nombres positifs : de « 1,401298E-45 » à « 3,4028235E+38 ». |
| DOUBLE | Contient des nombres signés IEEE 64 bits à virgule flottante de double précision (8 octets). Les nombres en double précision stockent une approximation d'un nombre réel. | « Do » | Pour les nombres négatifs : de « -1.79769313486231570E+308 » à « -4.94065645841246544E-324 ». Pour les nombres positifs : de « 4.94065645841246544E-324 » à « 1.79769313486231570E+308 ». |
| BYTE | Contient des entiers non signés de 8 bits (1 octet). | « By » | Toutes valeurs entières entre : « 0 » et « 255 ». |
| STRING | Représente une chaîne de texte constituée de caractères Unicode. | « St » | Tous les caractères UTF-8 à l'exception des caractères réservés « { », « } » et « : ». |
| NUL | Représente une donnée de valeur nulle. Ce type est utilisé en cas d'erreur. | « Nil » | La chaîne de caractère « Nil ». |

Tableau 1 : Type de données du protocole

2.1 Représentation des nombres

MarathonTP supporte les nombres sous la notation à point fixe ou scientifique. Par contre, le protocole ne définit pas de règle précise sur la représentation de l'une ou l'autre des notations. Les exemples suivants sont tous des représentations valides :

Exemples :

0.000135569887426

1.35569887426E-05

220000000000000000

2.2E17

2.2 Notes sur les chaînes de texte

MarathonTP utilise trois caractères UTF-8 spécifique dans son implémentation. L'utilisation de ces caractères dans une chaîne de texte occasionnera des exceptions au niveau des interpréteurs de message. Si ces caractères doivent malgré tout être transférés d'une application à l'autre un mécanisme doit être imaginé par le développeur.

| Caractère UTF-8 | Fonction |
|-----------------|-------------------------------|
| { (Hex 7B) | Marqueur de début de message. |
| } (Hex 7D) | Marqueur de fin de message. |
| : (Hex 3A) | Séparateur de champs. |

Tableau 2 : Caractères réservés

3 \\Structure de message

Le protocole MarathonTP définit un modèle de message précis permettant aux dispositifs compatibles d'échanger des informations entre eux.

Un message MarathonTP, aussi appelé « paquet », se compose de quatre éléments fondamentaux.

1. Les marqueurs de début et de fin de message
2. Les séparateurs de champs
3. Le descripteur
4. La charge utile



Figure 2 : Format d'un paquet MarathonTP

3.1 Définition des éléments

3.1.1 Les marqueurs de début et de fin

Les messages MarathonTP débutent par le caractère UTF-8 « { », Hex 7B et se terminent par le caractère UTF-8 « } » Hex 7D.

Il s'agit de deux caractères réservés par le protocole qui ne peuvent en aucun cas faire partie du message en lui-même. (Voir le point 2.2 pour la gestion d'exception)

3.1.2 Les séparateurs de champs

Les séparateurs de champs délimitent les différents éléments constitutifs d'un paquet. Ils ne sont présents que dans le descripteur et la charge utile. Ils sont représentés par le caractère UTF-8 « : », Hex 3A.

Il s'agit d'un caractère réservé par le protocole qui ne peut en aucun cas faire partie du message en lui-même. (Voir le point 2.2 pour la gestion d'exception)

3.1.3 Le descripteur

Le descripteur sert à identifier la nature du message. Il contient 4 éléments descriptifs obligatoires. Le tableau suivant fait l'énumération des ces éléments :

| Éléments | Description | Condition | Valeurs possibles |
|------------|---|-----------|--------------------|
| VER | Version du protocole. | Requis | « 1.0 » ou « 1.1 » |
| RA | Type de message « requête » ou « réponse ». | Requis | « R » ou « A » |
| TNS | Numéro de transaction. | Requis | « 0-65535 » |
| CMD | Commande du message. | Requis | « 0-255 » |

Tableau 3 : Champs de descripteur

VER : La version du protocole permet l'évolution de celui-ci. Ainsi, un système particulier pourrait répondre à l'une ou l'autre des versions disponibles selon les fonctions utilisées. Ce champ assure l'interopérabilité des systèmes.

RA : Le type de message. Pour chaque commande il y a deux types de message. La requête « R » et la réponse « A ».

TNS : La notion de transaction permet la traçabilité d'un message entre deux systèmes. Tous les messages actifs doivent impérativement avoir un numéro de transaction unique. Comme la quantité de numéro est limitée à 65536, il est possible de reprendre des numéros déjà utilisés. Pour autant qu'est respectée la règle d'unicité.

CMD : Il existe trois types de message MarathonTP. Ceux-ci sont traités au point 4.

3.1.4 La charge utile

La charge utile est le cœur du message. C'est à ce niveau que se retrouve toutes les données d'échange entre les machines. La charge utile se présente sous la forme d'une liste d'objet. Le format de ces objets varie selon la nature du message échangé. Il y a trois types d'objet. Les objets simples, doubles et triples qui contiennent respectivement 1, 2 et 3 éléments. La figure suivante illustre l'exemple d'une liste d'objets triples :

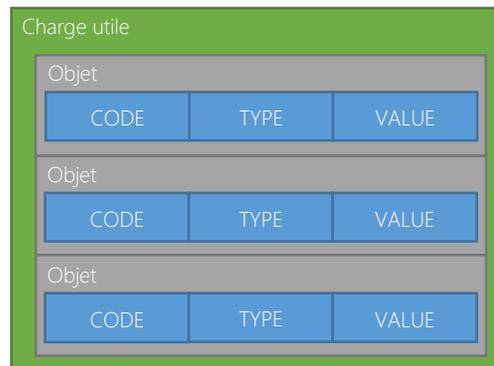


Figure 3 : Liste d'objets triples

4 \\Commandes MarathonTP

Le modèle transactionnel de MarathonTP utilise un code de commande pour une tâche bien précise à exécuter par le système qui la reçoit.

Comme il a été vu à la section 3.1.3, la commande de message est le 4^e paramètre du descripteur. Il s'agit d'un élément obligatoire. Le protocole est prévu pour définir jusqu'à 255 commandes différentes. À l'heure actuelle, MarathonTP définit trois commandes distinctes. La structure de message associé à chacune de ces commandes suit un format bien précis. La conformité d'une implémentation MarathonTP dépend grandement du respect de ce formatage.

Puisque le protocole est de type requête/réponse, on retrouve trois formats de requête et trois formats de réponse. Pour un total de 6 formats différents.

4.1 Lecture de variable

Il s'agit de la commande « 1 ». La lecture de variable permet d'obtenir la valeur d'un ou plusieurs éléments de la liste d'échange du système visé. La liste d'échange est traitée au point 8. La quantité maximale d'élément pouvant être lu avec la même requête est limité à 10. Combiner plusieurs éléments en une seule requête permet, dans de nombreux scénarios, de maximiser la bande passante du réseau sous-jacent.

4.1.1 Format de la requête

Le tableau suivant énumère les différents éléments de la requête.

| Éléments | Description | Valeurs possibles |
|----------|---|--------------------|
| VER | Version du protocole. | « 1.0 » ou « 1.1 » |
| RA | Type de message « requête » ou « réponse ». | « R » |
| TNS | Numéro de transaction. | « 0-65535 » |
| CMD | Commande du message. | « 1 » |
| ... | Début d'énumération. | |
| ELE | Élément de la liste d'échange. | « 0-65535 » |
| ... | Fin d'énumération. | |

Tableau 4 : Éléments d'une requête de lecture

Exemple de message :

| | | | | | |
|-----|----|-------|-----|-----|-----|
| 1.1 | R | 25693 | 1 | 0 | 1 |
| VER | RA | TNS | CMD | ELE | ELE |

Paquet : {1.1:R:25693:1:0:1}

4.1.2 Format de réponse

L'énumération de réponse doit correspondre à l'énumération de la requête correspondante. Chaque élément est accompagné d'un code de réponse tel que défini au point 4.4.

Pour tout code de réponse différent de 0, il est obligatoire d'utiliser le type d'élément « NUL » du tableau 1, accompagné d'une valeur de 0.

| Éléments | Description | Valeurs possibles |
|----------|---|--------------------|
| VER | Version du protocole. | « 1.0 » ou « 1.1 » |
| RA | Type de message « requête » ou « réponse ». | « A » |
| TNS | Numéro de transaction. | « 0-65535 » |
| CMD | Commande du message. | « 1 » |
| ... | Début d'énumération. | |
| CODE | Code de réponse. | Voir le tableau 10 |
| TYP | Type de l'élément | Voir le tableau 1 |
| VALUE | Valeur de l'élément | Voir le point 2 |
| ... | Fin d'énumération. | |

Tableau 5 : Éléments d'une réponse de lecture

Exemple 1 : Message résultant de la requête du point 4.1.1. :

| | | | | | | | | | |
|-----|----|-------|-----|------|------|-------|------|------|-----------|
| 1.1 | A | 25693 | 1 | 0 | Si | 84.83 | 0 | Do | 8.936E+10 |
| VER | RA | TNS | CMD | CODE | TYPE | VALUE | CODE | TYPE | VALUE |

Paquet : {1.1:A:25693:1:0:Si:84.83:0:Do:8.936E+10}

Exemple 2 : Message résultant de la requête du point 4.1.1, mais dont l'élément « 1 » ne fait pas partie de la liste d'échange du récepteur. :

| | | | | | | | | | |
|-----|----|-------|-----|------|------|-------|------|------|-------|
| 1.1 | A | 25693 | 1 | 0 | Si | 84.83 | 1 | Nil | 0 |
| VER | RA | TNS | CMD | CODE | TYPE | VALUE | CODE | TYPE | VALUE |

Paquet : {1.1:A:25693:1:0:Si:84.83:1:Nil:0}

4.2 Écriture de variable

Il s'agit de la commande « 2 ». L'écriture de variable permet de mettre à jour la valeur d'un ou plusieurs éléments de la liste d'échange du système visé. La liste d'échange est traitée au point 8. La quantité maximale d'élément pouvant être mis à jour avec la même requête est limité à 10. Combiner plusieurs éléments en une seule requête permet, dans de nombreux scénarios, de maximiser la bande passante du réseau sous-jacent.

4.2.1 Format de la requête

Le tableau suivant énumère les différents éléments de la requête.

| Éléments | Description | Valeurs possibles |
|----------|---|--------------------|
| VER | Version du protocole. | « 1.0 » ou « 1.1 » |
| RA | Type de message « requête » ou « réponse ». | « R » |
| TNS | Numéro de transaction. | « 0-65535 » |
| CMD | Commande du message. | « 2 » |
| ... | Début d'énumération. | |
| ELE | Élément de la liste d'échange. | « 0-65535 » |
| VALUE | Valeur à inscrire à l'élément. | Voir le point 2 |
| ... | Fin d'énumération. | |

Tableau 6 : Éléments d'une requête d'écriture

Exemple de message :

| | | | | | | | |
|-----|----|-------|-----|-----|-------|-----|------------|
| 1.1 | R | 25693 | 2 | 0 | 25.6 | 1 | 8.15698563 |
| VER | RA | TNS | CMD | ELE | VALUE | ELE | VALUE |

Paquet : {1.1:R:25693:2:0:25.6:1:8.15698563}

4.2.2 Format de la réponse

Une écriture de variable ne retourne pas de valeur. Simplement un code de réponse. L'ordre dans lequel sont envoyés les codes de réponse est le même que celui du message de requête correspondant.

| Éléments | Description | Valeurs possibles |
|----------|---|--------------------|
| VER | Version du protocole. | « 1.0 » ou « 1.1 » |
| RA | Type de message « requête » ou « réponse ». | « A » |
| TNS | Numéro de transaction. | « 0-65535 » |
| CMD | Commande du message. | « 2 » |
| ... | Début d'énumération. | |
| CODE | Code de réponse | Voir le tableau 10 |
| ... | Fin d'énumération. | |

Tableau 7 : Éléments d'une réponse d'écriture

Exemple de message résultant de la requête du point 4.2.1, mais dont l'élément « 1 » ne fait pas partie de la liste d'échange du récepteur. :

| | | | | | |
|-----|----|-------|-----|------|------|
| 1.1 | A | 25693 | 2 | 0 | 1 |
| VER | RA | TNS | CMD | CODE | CODE |

Paquet : {1.1:A:25693:2:0:1}

4.3 La Découverte

Il s'agit de la commande « 3 ». La Découverte sert à vérifier l'existence d'un système compatible MarathonTP sur le réseau de communication. La Découverte est une requête spéciale qui ne s'attache pas au processus d'envoi/réception standard tel que défini au point 5. Ainsi, le numéro de transaction n'est défini qu'à titre de compatibilité avec les autres commandes. La Découverte peut être particulièrement efficace en mode de diffusion réseau (broadcast). Ainsi, une requête unique devrait générer autant de réponse qu'il y a de systèmes MarathonTP sur le réseau. La Découverte utilise l'index 2 et 3 de la liste d'échange qui retourne l'identifiant IS du système ainsi que le mode de sécurité de ce dernier. Voir la section 7 pour plus d'information sur les codes de sécurité. La corrélation de la réponse avec la base de données ouverte d'Inertia Systèmes est laissée au système l'utilisant.

Note :

La Découverte est toujours utilisée sans aucun niveau de sécurité. De ce fait, en mode de diffusion réseau (broadcast), le protocole fixe à 5 secondes le délai minimal d'envoi d'une nouvelle requête, limitant ainsi la charge réseau.

4.3.1 Format de la requête

Le tableau suivant énumère les différents éléments de la requête de Découverte. L'élément « 2 » doit toujours précéder l'élément « 3 ».

| Éléments | Description | Valeurs possibles |
|----------|---|-------------------|
| VER | Version du protocole. | « 1.1 » |
| RA | Type de message « requête » ou « réponse ». | « R » |
| TNS | Numéro de transaction. | « 0-65535 » |
| CMD | Commande du message. | « 3 » |
| ... | Début d'énumération. | |
| ELE | Élément de la liste d'échange. | « 2 » ou « 3 » |
| ... | Fin d'énumération. | |

Tableau 8 : Éléments d'une requête de Découverte

Exemple de message :

| | | | | | |
|-----|----|-------|-----|-----|-----|
| 1.1 | R | 25693 | 3 | 2 | 3 |
| VER | RA | TNS | CMD | ELE | ELE |

Paquet : {1.1:R:25693:3:2:3}

4.3.2 Format de la réponse

L'ordre des valeurs de réponse de Découverte doit correspondre à l'énumération de la requête correspondante. Soit l'élément « 2 » suivi de l'élément « 3 » de la liste d'échange. Chaque élément est accompagné d'un code de réponse tel que défini au point 4.4.

Pour tout code de réponse différent de 0, il est obligatoire d'utiliser le type d'élément « NUL » du tableau 1, accompagné d'une valeur de 0.

| Éléments | Description | Valeurs possibles |
|----------|---|--------------------|
| VER | Version du protocole. | « 1.1 » |
| RA | Type de message « requête » ou « réponse ». | « A » |
| TNS | Numéro de transaction. | « 0-65535 » |
| CMD | Commande du message. | « 3 » |
| ... | Début d'énumération. | |
| CODE | Code de réponse. | Voir le tableau 10 |
| TYP | Type de l'élément | Voir le tableau 1 |
| VALUE | Valeur de l'élément | Voir le point 2 |
| ... | Fin d'énumération. | |

Tableau 9 : Éléments d'une réponse de Découverte

Exemple de message résultant de la requête du point 4.3.1 :

| | | | | | | | | | |
|-----|----|-------|-----|------|-----|--------------------------------------|------|-----|-------|
| 1.1 | A | 25693 | 3 | 0 | St | 76be3439-414b-4646-808d-af457aa6ddd6 | 0 | By | 0 |
| VER | RA | TNS | CMD | CODE | TYP | VALUE | CODE | TYP | VALUE |

Paquet : {1.1:A:25693:3:0:St:76be3439-414b-4646-808d-af457aa6ddd6:0:By:0}

4.4 Codes de réponse

Toutes les réponses MarathonTP sont accompagnées de code définissant l'état des transactions. Il y a des circonstances où les requêtes MarathonTP occasionneront des erreurs dans les systèmes qui les exécuteront. Le protocole prévoit quatre types d'erreurs définies selon un code numérique. Le tableau ci-dessous en fait l'énumération :

| Code d'erreur | Description |
|---------------|---|
| 0 | L'opération s'est correctement déroulée. |
| 1 | L'élément est introuvable. |
| 2 | Le type de donnée est incompatible. Cette erreur survient lors de l'écriture de variable. |
| 3 | Index en dehors des limites de la liste d'échange. |

Tableau 10 : Codes d'erreur MarathonTP

5 \\Organigramme des messages

Les messages MarathonTP sont de type requête/réponse. Cette méthodologie implique que l'envoi d'un message vers un destinataire doit impérativement se terminer par la réception de la réponse au niveau de l'émetteur du message. Le tout dans un délai défini par le contrôle de congestion. (Voir point 6) Si ces conditions ne sont pas rencontrées, alors le message est considéré perdu et doit être envoyé de nouveau au besoin.

Note : La Découverte est un type particulier de message qui ne requiert pas obligatoirement de réponse. Une requête et une réponse de Découverte doivent être traitées comme des événements uniques associés à une transaction factice.

5.1 Émission d'un message

L'organigramme suivant illustre la séquence d'opération pour l'émission d'un message de requête MarathonTP.

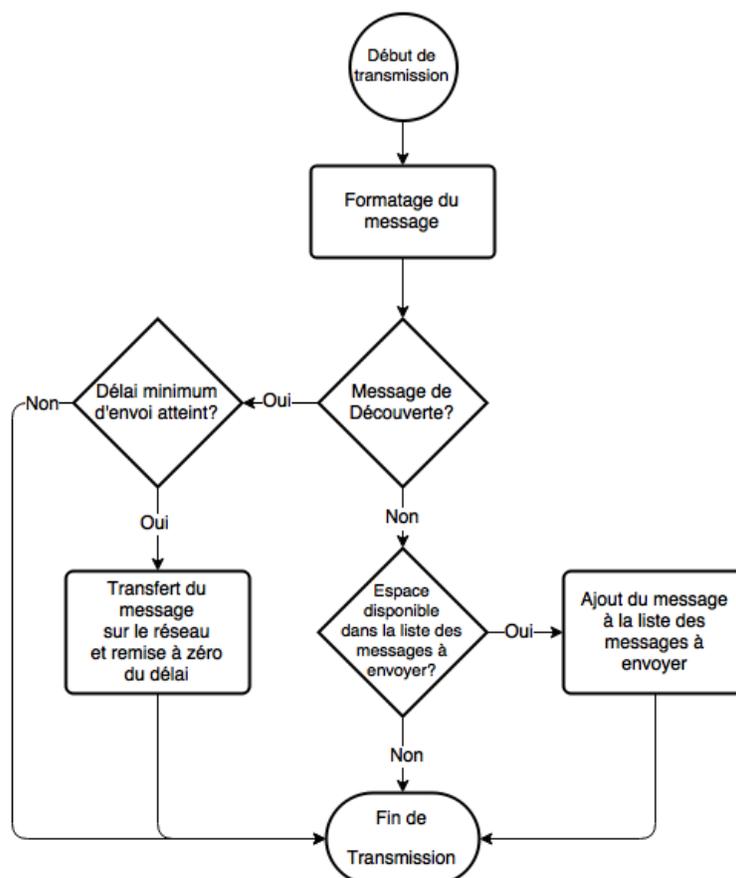


Figure 4 : Organigramme d'émission

5.2 Réception d'un message

L'organigramme suivant illustre la séquence d'opération de la réception des messages MarathonTP.

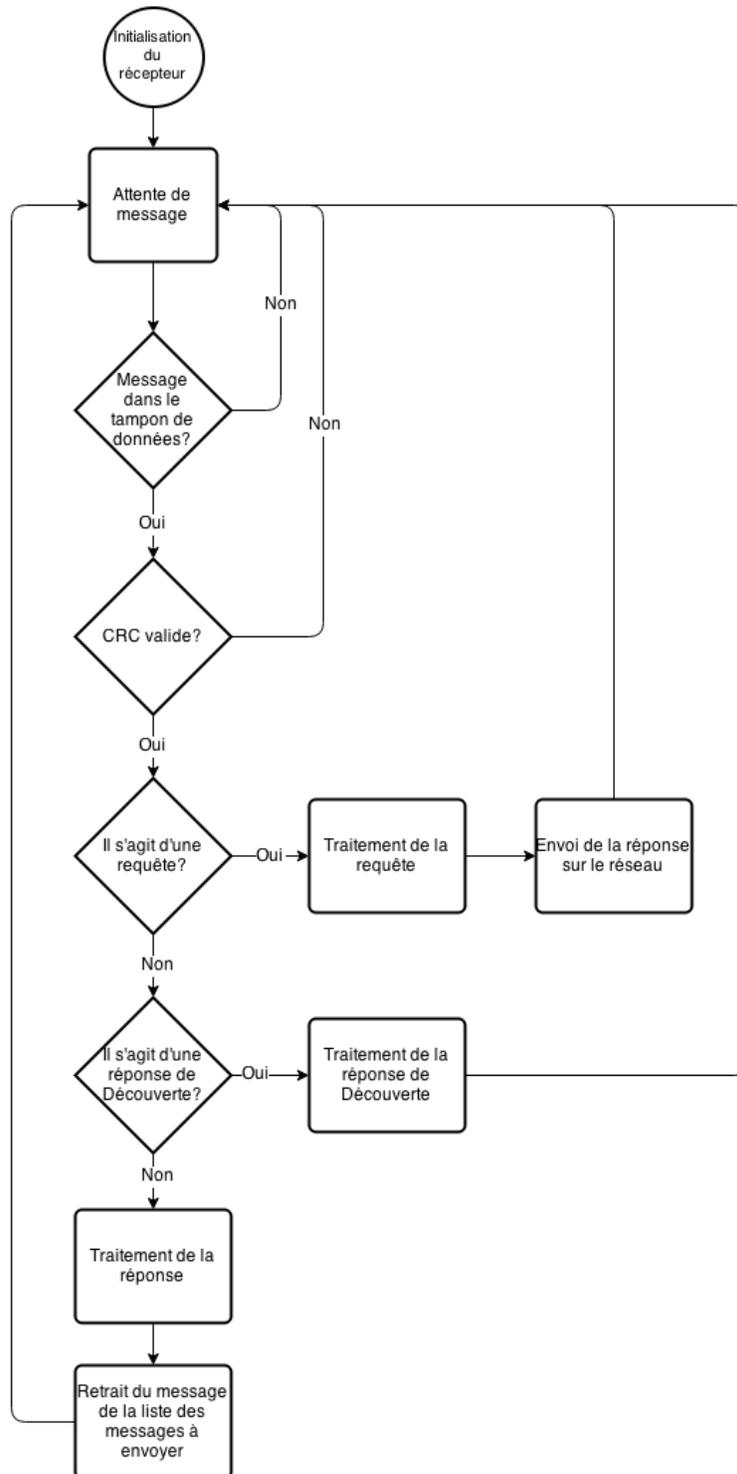


Figure 5 : Organigramme de réception

5.3 Recyclage des messages

L'organigramme suivant illustre la séquence d'opération d'un recycleur de message MarathonTP.

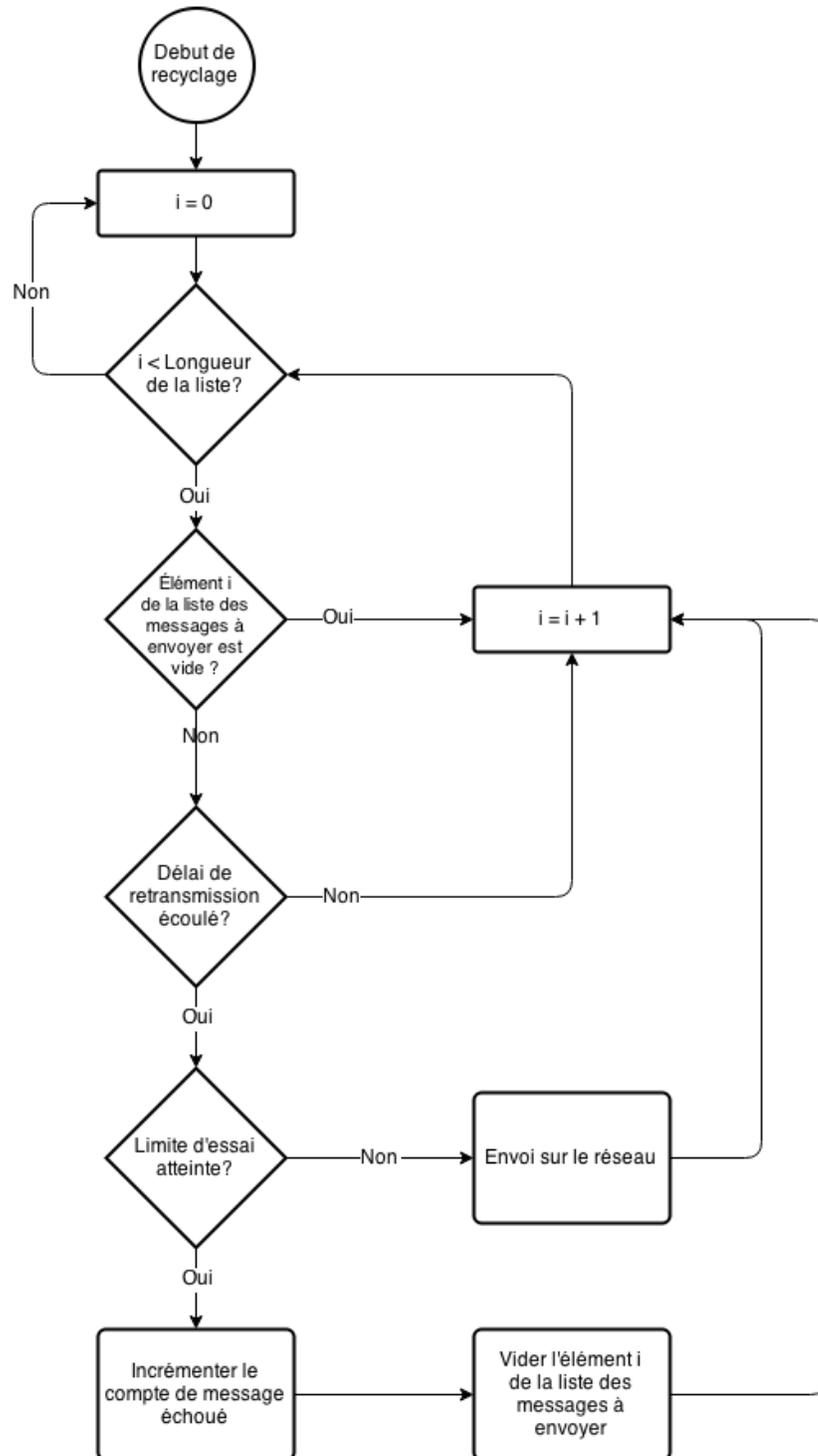


Figure 6 : Organigramme de recyclage

6 \Contrôle de congestion

Le fardeau du contrôle de congestion repose principalement sur les serveurs MarathonTP. De leur côté, les clients transmettent toujours les réponses selon une logique du meilleur effort. Autrement dit, aussitôt qu'une requête est traitée la réponse est immédiatement envoyée au serveur. Dans les situations où la charge réseau est grande, il est impératif d'introduire un contrôle des émissions de messages. Les mécanismes basiques utilisés à cette fin par MarathonTP sont le recul exponentiel et le régulateur d'envoi. Chacun d'eux ayant pour but de limiter les envois de messages afin d'éviter la surcharge du réseau.

6.1 Le régulateur d'envoi

Dans MarathonTP, le régulateur d'envoi permet de maximiser les capacités de transfert et de traitement des périphériques MarathonTP. En effet, dans un échange entre un client et un serveur, une partie du délai d'attente entre l'émission du message et la réception de la réponse est induit par le traitement des messages. Un régulateur d'envoi correctement configuré pourrait transmettre des messages à un serveur avec un rythme supérieur au temps de réponse moyen.

Le protocole prévoit une liste de message à envoyer. Il s'agit d'une pile contenant tous les messages en attente de transfert vers un destinataire spécifique. Il y a donc autant de pile de message que de destinataire actif au sein de l'application. Ce type d'attribution permet d'évaluer des scénarios de contrôle de la congestion adaptés à chaque destinataire. Ce type d'analyse prévient la détérioration des capacités d'envoi vers des dispositifs dont les temps de réponses sont bons. Si la pile est pleine, aucun nouveau message ne pourra être transmis sur le réseau. La taille de la liste est un facteur calculé par le régulateur d'envoi.

Note :

À l'heure actuelle, la spécification du protocole limite la liste de message à envoyer à un seul élément par destinataire. Une définition du régulateur d'envoi est à prévoir dans une spécification future.

6.2 Recul Exponentiel

Ce mécanisme vise à augmenter le délai d'attente « TimeOut » d'une réponse de message si ce délai a déjà été atteint. L'augmentation se fait par un facteur multiplicatif de 2 à chaque fois que le délai précédent est atteint. Deux paramètres influencent le fonctionnement de ce contrôle de congestion. Il s'agit du « Max Retransmit Interval » et du « Max Retry Attempt ». (Voir le point 8 pour la définition de ces paramètres). Dans tous les cas, un message ne pourra être émis de nouveau sur

le réseau si l'un ou l'autre des paramètres précédents est atteint. Dans un tel cas, le message est retiré de la liste des messages et la variable « Failed Message Count » est incrémentée.

Le protocole définit le délai d'attente initial à 3 secondes. Chacun des paramètres « TimeOut », « Max Retransmit Interval » et « Max Retry Attempt » sont modifiable par implémentation d'une telle fonctionnalité. Conformément à [RFC 5405](#), le « TimeOut » minimum est fixé à 1 seconde.

7 \Sécurité

Parce que MarathonTP a été conçu dans le but d'offrir un moyen de communication pour l'internet des choses (IoT), il est requis de prévoir un moyen de sécuriser les échanges d'information. Par contre, il est essentiel de considérer que la plupart des dispositifs IoT ont de faibles capacités de traitement. De ce fait, le protocole prévoit trois niveaux de sécurité. L'implémentation doit fournir un moyen de changer le mode de sécurité du dispositif l'utilisant. La réponse de Découverte (point 4.3.2) doit inclure une information spécifiant le mode de sécurité du dispositif interrogé. De cette façon, les clients peuvent ajuster leur communication en utilisant le mode adéquat. Le tableau suivant fait la synthèse des modes de sécurité que le protocole supporte :

| Mode de sécurité | Valeurs possibles |
|------------------|-------------------|
| AUCUNE | « 0 » |
| XTEA | « 1 » |
| AVANCÉE | « 2 » |

Tableau 11 : Modes de sécurité

7.1 Aucune Sécurité

Un dispositif placé dans ce mode échangera des messages sur le réseau avec tout autre tiers au moyen de message en texte brute non crypté. Il est de la responsabilité de l'utilisateur de bien comprendre ce qu'implique l'utilisation de ce type de transmission.

7.2 XTEA (eXtended TEA)

XTEA est un algorithme de chiffrement sécurisé qui utilise une clé de 128 bits et nécessite peu de puissance processeur. C'est un choix idéal dans le contexte de l'Internet des Objets (IoT). Il est important de souligner que ce chiffrement n'est pas aussi sécuritaire que des implémentations plus complexes comme RSA par exemple. Référez-vous au site web suivant pour des exemples d'implémentation XTEA. <http://marathontp.info>

7.3 Sécurité avancée

Aucune technologie de chiffrement avancée n'a encore été fixée pour le protocole. Cette fonctionnalité sera présentée dans une version future.

8 \\Listes d'échange

Les accès aux données dans MatahonTP se font par le biais d'index. Ces index sont de type « UShort ». Il y a donc 65536 données distinctes qui peuvent être transférée entre deux systèmes MarathonTP dont 65436 utilisable par les tiers.

Une certaine quantité de ces données sont réservé par le protocole. Le tableau suivant illustre la répartition des plages d'index.

| Index | Élément | Type | Description |
|-------------|-----------------------|---------|---|
| 0 | Ping | Boolean | Réponse Ping Marathon TP. |
| 1 | Device Serial | String | Numéro de série du dispositif. |
| 2 | Device IS Identifier | String | Numéro d'identification Inertia Systèmes du dispositif. |
| 3 à 9 | | | Réservé MarathonTP. |
| 10 | Sended Count | Integer | Compte de message envoyé. |
| 11 | Received Count | Integer | Compte de message reçu. |
| 12 | Failed Count | Integer | Compte de message en échec. |
| 13 | Retried Count | Integer | Compte de renvoi de message. |
| 14 | Successful Per Second | UShort | Compte de message réussi par seconde. |
| 15 à 99 | | | Réservé MarathonTP. |
| 100 à 65535 | | | Utilisable par les tiers. |

Tableau 12 : Répartition des plages d'index.

Il est du rôle du concepteur du dispositif de fournir un document précisant la fonction des 65436 index utilisables pour ses besoins. C'est ce document qui constitue la liste d'échange. La liste d'échange doit être accessible à tout utilisateur du dispositif.

8.1 Précision des index réservés

Les points suivants décrivent avec plus de précision la nature et la fonction des index réservés par le protocole.

8.1.1 Index 0 : Ping

Le « Ping » retourne simplement au client une réponse de type booléenne dont la valeur est vraie. La réception d'une réponse vraie par le client signale l'existence d'un serveur actif à l'adresse IP interrogée.

8.1.2 Index 1 : Device Serial

Le numéro de série unique du dispositif, présentée sous forme textuelle. Le numéro de série est propre au fabricant du dispositif.

8.1.3 Index 2 : Device IS Identifier

Identificateur unique de dispositif fourni par Inertia Systèmes lors de l'approbation de la conformité MarathonTP. Cet identificateur textuel permet aux services et logiciels de consulter la base de données de dispositif d'Inertia Systèmes et d'obtenir les caractéristiques du produit.

8.1.4 Index 3 : Security Mode

Nombre entre 0 et 2 inclusivement, représentant le mode de sécurité du dispositif interrogé.

8.1.5 Index 10 : Sended Count

Nombre incrémental représentant la quantité de message envoyé par le dispositif. Ce paramètre est utile à des fins d'analyse et de statistique. Le dispositif doit gérer le retour à 0 lors du dépassement de capacité d'un « Integer ».

8.1.6 Index 11 : Received Count

Nombre incrémental représentant la quantité de message reçu par le dispositif. Ce paramètre est utile à des fins d'analyse et de statistique. Le dispositif doit gérer le retour à 0 lors du dépassement de capacité d'un « Integer ».

8.1.7 Index 12 : Failed Count

Nombre incrémental représentant la quantité de message qui n'ont jamais été interprété par le receveur. Ce paramètre est utile à des fins d'analyse et de statistique. Le dispositif doit gérer le retour à 0 lors du dépassement de capacité d'un « Integer ».

8.1.8 Index 13 : Retried Count

Nombre incrémental représentant la quantité de renvoi de message par le dispositif. Ce paramètre est utile à des fins d'analyse et de statistique. Le dispositif doit gérer le retour à 0 lors du dépassement de capacité d'un « Integer ».

8.1.9 Index 14 : Successful Per Second

Nombre représentant la quantité de message correctement échangés entre deux dispositifs MarathonTP à chaque seconde. Ce nombre est mis à jour aux secondes. Ce paramètre est utile à des fins d'analyse et de statistique.

8.1.10 Index 15 : Max Retransmit Interval

Nombre en milliseconde représentant le délai maximal absolu pour l'émission d'un message.

8.1.11 Index 16 : Max Retry Attempt

Nombre de nouvelle tentative permise.

8.1.12 Index 17 : TimeOut

Nombre en milliseconde représentant le délai permit entre l'émission d'un message et la réception d'une réponse.

9 \\Conformité du protocole

Pour assurer la compatibilité du protocole MarathonTP entre des systèmes de technologies variée, certaines caractéristiques fondamentales doivent être respectées.

9.1 Port de communication UDP

MarathonTP possède un port de communication UDP/IP réservé. Il s'agit du port **8384**.

9.2 Types de données

Au minimum, tous les types de données du tableau 1 doivent être disponibles.

9.3 Taille des paquets

Bien que MarathonTP limite à 10 le nombre d'élément par paquet, il reste de la responsabilité du développeur de respecter les contraintes du protocole de transport utilisé. Les paquets ne doivent pas être scindés.